

APPENDIX

APIs

Workload Model APIs

❖ **pti_workload**

Void	Define(pti_evnameid, DOD)	Specify a new event / workload using the event list name ID
Void	Define(pti_evlist*,DOD)	Specify a new event / workload using a pointer to the event list
Void	Eval(void)	Evaluate all existing lists
Void	Eval(pti_elmark, pti_elmark)	Evaluate a part of the event list specifying start and end
Void	Eval(pti_elmark, pti_elmark, int, pti_evalalg)	Evaluate a part of the event list specifying start and end many times using the suggested evaluation algorithm
pti_evalres	FlushResults	Return overall result of evaluation
Void	FlushOtraces	Create output traces

5

❖ **pti_ddpar**

Void	Add(pti_parref)	Add a new data dependency parameter
Void	SetValue(pti_parref*,void*)	Set value to parameter
pti_parref*	Search(string)	Search dd parameter by name
void*	GetValue(pti_parref*)	Get current value of dd parameter

❖ **pti_register**

Void	Add(pti_mtype,void*, Schema)	Add a new model to the system by defining it's type, pointer to its class, and related XML schema
Void	Delete(pti_mref)	Delete model from system
pti_mref	Search(pti_type,string)	Search a specific type of model by name
pti_mref	Search(string)	Search all model by name

Hardware Model APIs

❖ **pti_hmod**

Void	RegInit	Initialise hardware model (when registered)
String	GetName	Name of model
pti_parref	GetParConfig	Get model configuration parameters
Schema	GetParSchema	Get model configuration XML schema for HMC DB
pti_hmdev	GetDeviceType	Device type (Part of which device)
pti_hmscope	GetScope	Scope of model (single or multiple events)
pti_mref	GetWorkloadType	Workload type processed by the model
pti_mref	GetOtraceType	Extended trace output created by evaluation
Void	Eval(pti_event)	Evaluate hardware model – for SEVENT scope
Void	Eval(pti_elmark, pti_elmark)	Evaluate hardware model – for MEVENT scope

❖ **pti_event**

DOD	GetWorkload	Returns workload XML script
void	SetEvalRes(pti_evalres)	Hardware model sets the result of evaluation
void	SetOtrace(DOD)	Hardware model sets extended output trace as a result to evaluation
void	AddEvent(pti_evlist* pti_evtype,DOD, pti_event*[])	Add new event at specified event list, specify type, XML workload, and pointers to related events (from and to).

❖ **pti_eviter**

pti_event*	FirstEvent(pti_evlist*)	Access first element of the list
pti_event*	LastEvent(pti_evlist*)	Access last element of the list
pti_event*	NextEvent(pti_evlist*)	Access next element of the list
pti_event*	PreviousEvent(pti_evlist*)	Access previous element of the list

❖ **pti_accevlis**

pti_evlist*	Search(pti_evlnameid)	Searches for an event list with a specified name ID*
pti_evlist*	First(void)	First event list
pti_evlist*	Next(void)	Next event list
pti_evlist*	Previous(void)	Previous event list
pti_evlist*	Last(void)	Last event list

Hardware & Model Configuration APIs

❖ **pti_hmc**

Void	Config(DOD)	Configure engine based on DOD configuration
------	-------------	---

Output Trace APIs

❖ **pti_otrace**

Void	SetFilter(void (*f)())	Specify filter function for otraces
Void	SetFile(string)	Specify output filename
Void	SetMode(pti_otmode)	Specify output trace mode

Data Types

- ❖ **Schema - Represents the XML schema**
Details on the actual type depend on the schema definition language.
- ❖ **DOD – Reference to XML Document Object Model**
- ❖ **pti_mtype – Type of model**
Enumerated values: HRDMOD, WRKMOD, OTMOD
- ❖ **pti_hctype – Type of hardware model configuration parameter**
Enumerated values: INT, FLOAT, STRING
- ❖ **pti_hmdev – Hardware device type**
Enumerated values: COMPUTER, NETWORK
- ❖ **pti_hmscope – Scope of hardware model**
SEVENT model process single event
MEVENT model process multiple events
- ❖ **pti_mref – Reference to any model type**
pti_mtype model type
void* pointer to model access class
Schema Schema to related XML DB
- ❖ **pti_parref – Generic parameter reference (used for hardware model configuration and data dependency parameters)**
string Name of parameters
pti_hctype Type of parameter
void* Pointer to hrdmod variable
- ❖ **pti_elmark – Mark specific positions of the event list (start or end of events)**
Not accessed directly through APIs (accessed through pti_hmeliter defined at the hardware model API)
- ❖ **pti_evalres – The result of the evaluation of the hardware model for a single event**
ulong predicted time
ulong best case predicted tx
ulong worst case predicted tx

❖ **pti_evtype – Type of event**

PROCESS	Processing or single event
COMM	Communication event (synchronous)
ACOMM	Asynchronous communication event
SYNCH	Synchronization event
WAIT	Wait for synchronization event

❖ **pti_evlist – A single event list**

The event list is the main data structure used by the evaluation engine to represent a sequence of events that takes place for a modeled system, store the results of individual models, and combine these results into the overall system performance prediction.

An event list is a single linked list representing events that take place on one of the system components (e.g. a single CPU or process). The event list is identified with the name id of the system component that it models. The elements of the list represent individual events. An event can be, by way of example, a computation, an I/O operation, a communication between computers, etc. Event interactions (e.g. process communications) are represented by pointers that may originate from the events that created the interactions and point to the target event(s).

❖ **pti_evlnameid – Name id is a string that determines one or more event lists**

The definition of an event list id is determined upon configuration of the system architecture at the HMC. Consider the following configuration script:

```
<system name="pc_cluster">
<computer name="pc_node" count="16">
  <!-- CPU Model C Operations -->
    Processes = 10
    Threads = 2
    <cpu_clc>
      <DILG>0.043</DILG>
      <IADD>0.127</IADD>
      <!-- Other operation follow -->
    </cpu_clc>
  </computer>
<network name="myrinet">
  <ccmod>
    <Nproc>16</Nproc>
    <-- other configuration follow -->
  </ccmod>
</network>

<connect>
  <computer name="pc_node" node="1">
    <network name="myrinet" port="1">
  </connect>
<!-- Connection to other nodes ... -->
</system>
```

To refer to all pc nodes the event list name ID is "PC_NODE". To refer to one node "PC_NODE.1". To refer to specific thread "PC_NODE.1.9.1". To refer to the network "MYRINET". Note: only the first part of the name ID is symbolic.

The user may, if desired, define more than one system component by omitting a level of the ID description. For example "PC_NODE.2" refers to all processes and threads of PC node 2.

❖ **pti_evalalg – Event list evaluation algorithm**

SIMULATION

Is extendable to many algorithms

5 ❖ **pti_otmode – Define output trace modes**

BASIC Output basic output traces (dump of event list)

EXTENDED Output hardware model extended traces

METRICS Use filtering process to produce metrics

10

APPENDIX END